



Personnaliser vos images ISO Linux

Benjamin.Barras@epfl.ch

EPFL – Domaine IT – Responsable de la plate-forme Linux à l'EPFL

The aim of this paper is to show that a small ISO image (8 to 12 Mbytes) is more than enough to install a Linux distribution and to show how we can customize our installation, by setting some parameters. This will be particularly useful for people using virtualization and often needing an ISO image. More generally, it is possible to use these settings regardless of the used boot method (ISO, USB, PXE or disk).

Le but de cet article est de montrer qu'une petite image ISO¹ (8 à 12 Mbytes) suffit largement pour installer sa distribution Linux et surtout de voir comment, en passant quelques paramètres, on peut personnaliser son installation.

Cela sera surtout très utile aux personnes qui font de la virtualisation et qui ont souvent besoin d'une image ISO. Plus généralement, il est possible d'utiliser ces paramètres quelle que soit la méthode d'amorçage utilisée (ISO, USB, PXE ou disque).

Introduction

Les personnes qui souhaitent installer Linux utilisent le plus souvent une image ISO au format DVD qui contient tout et facilite grandement l'installation. Le problème est que cette image est généralement obèse et pèse environ 3 à 4 Gbytes; elle occupe de la place sur notre espace de stockage, elle est longue à déplacer et, pour ceux qui font de la virtualisation, cela leur fait perdre du temps uniquement pour charger cette image. Si nous procédons de la sorte, c'est le plus souvent par méconnaissance des différentes méthodes d'installation des distributions Linux, voir par manque de temps ou simplement pour se simplifier la vie. Or, il existe une manière beaucoup plus légère pour installer sa distribution Linux, mais qui demande un petit peu plus de réflexion afin de bien préparer son amorce d'installation. Malheureusement, dans notre monde moderne, il y a de plus en plus une mode du clic qui peut paraître plus simple en apparence, mais qui devient très vite un enfer pour les administrateurs qui doivent installer leur distribution Linux sur plus d'une dizaine d'ordinateurs. La mode du clic est en passe de devenir la règle car toute réflexion en est généralement absente, et il devient de plus en plus difficile de s'en extraire. Mais un malheur n'arrivant jamais seul, un autre phéno-

mène a fait son apparition dans notre société et se dissimule sous le qualificatif d'obésité. Ce qui veut dire que la petite image ISO dont je parle ci-dessus, a malheureusement tendance à prendre du poids au fil des ans et ce qui est vrai aujourd'hui ne le sera pas forcément demain. Nous verrons qu'une petite image ISO, qui nous servira d'amorce, suffit largement à atteindre notre objectif mais il nous faudra, bien entendu, disposer d'un bon réseau. Par la suite, nous verrons comment utiliser le contenu de notre image pour le passer sur une clef USB de manière très simple. Un des buts de cet article est de dévoiler ce qui se cache derrière ces fameux clics et de bien comprendre comment préparer son amorce qui par la suite peut être déplacée sur un support différent (PXE, USB, disque, ...) car la base reste toujours la même, quelle que soit la distribution choisie, et c'est au nom de ce dénominateur commun que je convie notre lecteur.

Trouver son image

C'est là que les choses se corsent, car chaque distribution Linux a son propre système d'archivage. Mais on finira toujours par trouver une image ISO minimale pour beaucoup de ces distributions, même pour Ubuntu où pourtant le clic est le modèle de base. Pour Fedora Et Red Hat Enterprise Linux, on trouve une image ISO dans le dossier images de notre distribution, elle a pour nom **boot.iso**. Pour Ubuntu, on trouvera cette image ISO dans le dossier **current/images/netboot** du dossier **main/installer-amd64** ou **main/installer-i386** de notre distribution et aura pour nom **mini.iso**. Il n'existe pas de mode d'emploi, il faut juste utiliser son flair et comme notre installation va se faire par le réseau, il nous faudra également aiguïser notre patience afin de dénicher le bon dépôt.

Remarques

Ce qu'il faut savoir, c'est que la petite image ISO récupérée est souvent liée à une distribution et une version donnée de celle-ci. On ne peut pas, en règle générale, installer une version différente avec l'image récupérée. Mais nous verrons comment résoudre cette difficulté supplémentaire de manière très simple. D'autre part, tout ce qui s'applique à Red Hat s'applique également à CentOS, qui est construite à partir des sources d'un éminent fournisseur nord-américain de Linux Enterprise, et est librement accessible mais sans support.

¹ Je me suis concentré sur une image ISO, car je pense que c'est la méthode la plus simple à mettre en œuvre et cela nous servira de fil conducteur pour attaquer d'autres méthodes.

Premier exemple: à la Red Hat

Prenons pour commencer un exemple simple, soit la petite image **boot.iso** (voir annexe A.1.2) d'une Red Hat (RHEL Workstation 5.5, i386) et faisons ensuite quelques petites manipulations afin d'en récupérer son contenu:

```
# Répertoire de montage & monter l'image pour
#récupérer les fichiers
mkdir iso
sudo mount -o loop boot.iso ./iso
# On récupère les fichiers qui se trouvent dans
#le dossier isolinux
mkdir myIso
cp iso/isolinux/* ./myIso
# On se donne les droits d'écriture
chmod 644 myIso/isolinux.*
```

code 1

Le fichier important pour notre travail sera `isolinux.cfg` dont la documentation complète est référencée en annexe (A.3), ainsi que la liste complète des paramètres pouvant figurer dans ce fichier et les valeurs que peuvent prendre ces derniers. Il a typiquement la forme suivante:

```
DEFAULT linux
PROMPT 1
TIMEOUT 600
# ...
SAY Type linux for install linux
SAY Type text for install linux in text mode
LABEL linux
    KERNEL vmlinuz
    APPEND initrd=initrd.img
LABEL text
    KERNEL vmlinuz
    APPEND initrd=initrd.img text
```

code 2

où `DEFAULT` correspond au `LABEL` qui lancera l'installation par défaut, ici `linux`, et qui démarrera au bout de 60 secondes si l'on ne fait rien. `PROMPT 1` affichera une invite de commande (prompt) du style: **boot:** qui vous permettra pendant 60 secondes (`TIMEOUT` en 1/10 s) de choisir les options d'installation à disposition. On choisira l'option souhaitée en tapant le nom du label correspondant et on lancera l'installation avec la touche `ENTER`.

```
SAY Type epfl_dhcp_nfs for install with dhcp and nfs
SAY Type epfl_dhcp_http for install with dhcp and http
SAY Type epfl_ipFixe_nfs for install with fixed ip and nfs
SAY Type epfl_ipFixe_http for install with fixed ip and http
LABEL epfl_dhcp_nfs
    KERNEL vmlinuz
    APPEND initrd=initrd.img ip=dhcp method=nfs:linuxline.epfl.ch:/export/RHE/ws-5/update5/32-bit-x86/i386
LABEL epfl_dhcp_http
    KERNEL vmlinuz
    APPEND initrd=initrd.img ip=dhcp method=http://linuxline.epfl.ch/RHEL/ws-5/update5/32-bit-x86/i386
# Remplacer 128.178.x.y avec une adresse IP valable, x étant votre subnet
LABEL epfl_ipFixe_nfs
    KERNEL vmlinuz
    # ATTENTION ! Les paramètres doivent être sur une seule ligne
    APPEND initrd=initrd.img ip=128.178.x.y netmask=255.255.255.0 gateway=128.178.x.1
    dns=128.178.15.8,128.178.15.7 method=nfs:linuxline.epfl.ch:/export/RHE/ws-5/update5/32-bit-x86/i386
LABEL epfl_ipFixe_http
    KERNEL vmlinuz
    # ATTENTION ! Les paramètres doivent être sur une seule ligne
    APPEND initrd=initrd.img ip=128.178.x.y netmask=255.255.255.0 gateway=128.178.x.1
    dns=128.178.15.8,128.178.15.7 method=http://linuxline.epfl.ch/RHEL/ws-5/update5/32-bit-x86/i386
```

code 3

Il nous faudra juste afficher un petit menu, par exemple avec le mot-clé **SAY**, qui nous permettra de voir les différentes options à disposition. Chaque `LABEL` correspond à une tâche spécifique et ce sera à ce niveau que nous allons personnaliser notre installation en y ajoutant différents paramètres après le mot-clé **APPEND** qui seront ensuite passés à l'installateur Linux, soit *Anaconda* pour *Red Hat & Fedora*, et *Ubiquity* pour *Ubuntu*, qui dérive lui-même du *Debian-Installer*. Toute la difficulté consiste à trouver la documentation correspondant à ce que l'on appelle les *Boot Options*, qui sont en fait des paramètres que l'on passe soit au noyau, soit à l'installateur (A.2). Le mot-clé **KERNEL** indique le nom du noyau à charger et **APPEND** les options qui seront rajoutées lors de l'exécution du noyau, dont l'image d'un système minimal `initrd` ainsi que les différents paramètres que l'on souhaite passer à l'installateur. On ne se préoccupera ici, que des paramètres utiles pour l'installateur et pas des paramètres pouvant être passés au noyau. Les autres mots-clés figurant dans ce fichier ne concernent que l'esthétique du menu affiché.

Pour notre premier exemple, notre installateur sera *Anaconda* dont la liste complète des *Boot Options* est bien documentée. Comme notre image ISO est petite, la première chose que va faire notre installateur est d'aller chercher par le réseau les paquets nécessaires à l'installation de Linux. Pour ce faire nous avons besoin d'une adresse IP et d'un dépôt (*repository*). Il y a deux cas distincts, celui où nous avons une adresse IP fixe et l'autre où cette dernière est fournie par un serveur DHCP.

Nous pouvons éditer le fichier `isolinux.cfg` et ajouter les quelques lignes qui se trouvent ci-dessous (code 3).

Trois remarques

- Tous les paramètres derrière **APPEND** doivent être sur une et une seule ligne (la présentation ne permet pas de le faire).
- Le paramètre **method** est obsolète, il faudrait utiliser **repo** à la place, mais malheureusement Red Hat ne l'utilise pas encore. Fedora par contre, l'utilise abondamment et je ne parle pas d'Ubuntu qui utilise la syntaxe propre à Debian.
- Le but est d'avoir un label aussi simple que possible. Dans l'exemple ci-dessous les labels ont un but purement descriptif (code 3).

Personnaliser vos images ISO Linux

```
# Créer la nouvelle image ISO
mkisofs -o myBoot.iso -b isolinux.bin -c boot.cat \
        -no-emul-boot -boot-load-size 4 -boot-info-table myIso
```

code 4

```
DEFAULT epfl
PROMPT 1
TIMEOUT 600
SAY Type epfl_dhcp for install with dhcp
SAY Type epfl_ipFixe for install with IP fixed
LABEL epfl_dhcp
    KERNEL linux
    # ATTENTION ! Les paramètres doivent être sur une seule ligne
    APPEND mirror/protocol=http mirror/http/hostname=linuxline.epfl.ch
    mirror/http/directory=/ubuntu mirror/country=manual initrd=initrd.gz
LABEL epfl_ipFixe
    KERNEL linux
    # Il faut remplacer l'adresse IP, le hostname et le gateway en conséquence
    # ATTENTION ! Les paramètres doivent être sur une seule ligne APPEND netcfg/disable_dhcp=true
    netcfg/get_ipaddress=128.178.1.55 netcfg/get_nameservers=128.178.15.8
    netcfg/get_netmask=255.255.255.0 netcfg/get_gateway=128.178.1.1 netcfg/confirm_static=true
    netcfg/get_hostname=ditsbpc13 netcfg/get_domain=epfl.ch mirror/http/hostname=linuxline.epfl.ch
    mirror/protocol=http mirror/http/directory=/ubuntu mirror/country=manual initrd=initrd.gz
```

code 5

On remarquera ici que l'on peut aller chercher nos paquets d'installation via *nfs* ou *http*, ce sont les deux méthodes les plus courantes que je vous recommande, mais rien ne vous empêche de choisir une autre méthode en fonction des disponibilités de votre installateur et des services offerts par le serveur qui contient le dépôt. Il ne nous reste plus qu'à construire notre nouvelle image ISO comme le montre le code 4.

Attention, par défaut les noms de fichiers sont au format ISO 9660 et donc 8 caractères maximum pour le nom, 3 pour l'extension et pas d'extension pour les noms de répertoires, probablement dû à l'héritage du MS-DOS¹. Si vous devez avoir des noms de fichiers plus long, vous pouvez essayer de rajouter le paramètre `-iso-level 2` pour *mkisofs*, mais vous remarquerez que les noms par défaut de votre distribution Linux respectent le format standard.

Ensuite, il suffit de faire démarrer cette image ISO qui se nomme maintenant *myBoot.iso*, soit en la gravant, soit en l'utilisant dans une machine virtuelle. Nous nous retrouverons avec une installation identique à celle du DVD, quoique peut-être en mode texte, mais en prime bien plus rapide et de taille minimale.

Deuxième exemple: Ubuntu

Pour notre deuxième exemple, nous prendrons la petite image *mini.iso* d'Ubuntu (LTS 10.04) et ferons des manipulations similaires à l'exemple précédent, afin de récupérer le contenu de l'image ISO et de modifier le fichier *isolinux.cfg* comme montré dans le code 5.

La première chose qui nous frappe, c'est le nom des paramètres de boot qui devient long et peu intuitif. Cela est dû à l'installateur Debian qui est du type Question/Réponse, et que les utilisateurs Debian connaissent probablement sous la désignation technique de *preseed*. Malheureusement ici, la documentation est fortement

incomplète et le seul moyen de trouver tous les paramètres est de regarder le fichier *example-preseed.txt* cité dans la documentation Ubuntu (A.1.1), qui fort heureusement décrit également les limitations importantes de cet installateur lorsque l'on veut passer à une installation automatisée. De plus, les essais sont nécessaires car le comportement de l'installateur est quelque peu surprenant et le résultat final peut changer (démonstration à l'appui) en fonction de la méthode utilisée, contrairement à Anaconda qui est d'une fiabilité redoutable.

Dénominateur commun

Vous l'avez peut-être déjà compris, le point commun à toute ces installations, ce sont trois mots-clés: **kernel**, **initrd**, **ISOLINUX**. Soit un noyau Linux (**kernel**), une image d'un système minimal (**initrd**) utilisé au démarrage du noyau Linux et un outil pour faire démarrer notre image ISO (**ISOLINUX**). Avec ces trois ingrédients, on peut installer à peu près n'importe quelle distribution Linux. Bien sûr, il n'y a pas que ISOLINUX pour lancer notre noyau, on peut aussi utiliser GRUB, mais ISOLINUX reste d'une simplicité extrême et ses différentes variantes SYSLINUX, PXELINUX, EXTLINUX permettent de faire démarrer notre installation non seulement depuis une image ISO, mais aussi depuis une clef USB, ou un disque local au format ext2/ext3 ou le réseau. D'ailleurs, et ce n'est pas un hasard, pour chaque distribution Linux nous trouverons un dossier qui ne contiendra en général que le noyau et l'image d'un système minimal. On trouvera par exemple pour Ubuntu les deux fichiers Linux & *initrd.gz* dans le dossier *netboot*, pour RHEL & FEDORA ce sera les fichiers *vmlinux* & *initrd.img* dans le dossier *isolinux* et pour conclure avec openSUSE, les fichiers seront *linux* & *initrd* dans le dossier *loader*. Rien ne nous empêche d'embarquer l'esthétique avec, en la prenant par exemple sur l'image ISO et qui sera généralement au format ISOLINUX.

¹ Voir fr.wikipedia.org/wiki/MS-DOS: système rudimentaire, monotâche et monoutilisateur ...

```
DEFAULT U
PROMPT 1
TIMEOUT 600
SAY You need dhcp for this example !
SAY Type R1 for install RHEL 5.5 (workstation, i386)
SAY Type R2 for install RHEL 5.5 (workstation, 64-bit)
SAY ...
SAY Type F (maj.) for install Fedora 13 (i386)
SAY Type S (maj.)for install openSUSE 11.3 (workstation, i386)
SAY Type U (maj.)for install Ubuntu 10.04 LTS (i386)
LABEL R1
  KERNEL rhel55_1/vmlinuz
  # ATTENTION ! Les paramètres doivent être sur une seule ligne
  APPEND initrd=rhel55_1/initrd.img ip=dhcp
  method=nfs:linuxline.epfl.ch:/export/RHE/ws-5/update5/32-bit-x86/i386
LABEL R2
  KERNEL rhel55_2/vmlinuz
  # ATTENTION ! Les paramètres doivent être sur une seule ligne
  APPEND initrd=rhel55_2/initrd.img ip=dhcp
  method=nfs:linuxline.epfl.ch:/export/RHE/ws-5/update5/64-bit-x86/AMD64_iEM64T
#...
LABEL F
  KERNEL fedora/vmlinuz
  # Remarque: on utilise ici repo à la place de method
  APPEND initrd=fedora/initrd.img ip=dhcp repo=http://linuxline.epfl.ch/Fedora/f13/Fedora/i386/os
LABEL S
  KERNEL openSUSE/linux
  # ATTENTION ! Les paramètres doivent être sur une seule ligne
  APPEND initrd=openSUSE/initrd hostip=dhcp
  install=http://mirror.switch.ch/ftp/mirror/opensuse/distribution/11.3/repo/oss
LABEL U
  KERNEL ubuntu/linux
  # ATTENTION ! Les paramètres doivent être sur une seule ligne
  APPEND mirror/protocol=http mirror/http/hostname=linuxline.epfl.ch mirror/http/directory=/ubuntu
  mirror/country=manual initrd=ubuntu/initrd.gz
```

code 6

Dernier exemple: personnalisé

Pour notre dernier exemple, on va récupérer les deux premiers ingrédients (kernel & initrd), dans les différents dossiers indiqués précédemment, mais pour différentes distributions Linux et versions diverses. On va simplement glisser ces deux fichiers dans un répertoire spécialement dédié, ceci afin d'éviter les collisions de noms entre les différentes versions et distributions. C'est ce répertoire qui va nous permettre de différencier les différentes versions et distributions que l'on peut rajouter dans notre image. Il ne nous reste plus qu'à récupérer le fichier isolinux.bin d'un des précédent exemple et de le mettre à la racine de notre répertoire. On construira notre fichier isolinux.cfg comme le montre le code 6.

Vous pouvez récupérer l'image ISO de cet exemple (A.5) afin de tester et d'installer l'une des distribution ci-dessus. Cela vous permettra de comprendre la structure de cette image, et de passer de la théorie à la pratique. Si vous n'avez pas de serveur DHCP, vous pouvez modifier le fichier isolinux.cfg de cette image afin d'y mettre votre propre adresse IP fixe comme indiqué dans les exemples précédents.

Conclusion

On l'aura vite compris, ces exemples sont là dans un but démonstratif afin d'apprendre à bien préparer son amorce de base pour installer sa distribution Linux.

Maintenant si l'on veut aller plus loin, il faudra passer par un fichier qui contiendra tous les paramètres utiles à notre installation (la langue, les logiciels à installer, etc.). Ce sera un fichier de type KickStart pour RHEL, Fedora & Ubuntu (Ubuntu a aussi une autre option avec un fichier `preseed`).

Maintenant que nous avons tenté de démystifier l'amorce, nous pourrions nous pencher sur une installation plus avancée, voir complètement automatisée, dans de prochains articles.

Annexes

Vous trouverez ici une liste de quelques liens *périssables* , mais les mots-clés se trouvant à gauche du lien ne le sont eux généralement pas:

Personnaliser vos images ISO Linux

A.1 Où trouver nos distributions Linux ?

A.1.1 Ubuntu 10.04 LTS

i386: linuxline.epfl.ch/ubuntu/dists/lucid/main/installer-i386/current/images/netboot/mini.iso

x86_64: linuxline.epfl.ch/ubuntu/dists/lucid/main/installer-amd64/current/images/netboot/mini.iso

Preconfiguration file: <https://help.ubuntu.com/10.04/installation-guide/example-preseed.txt>

Preseed limitations: <https://help.ubuntu.com/7.04/installation-guide/i386/preseed-intro.html#preseed-limitations>

A.1.2 RHEL 5.5

Workstation i386: linuxline.epfl.ch/RHEL/ws-5/update5/32-bit-x86/i386/images/boot.iso

x86_64: linuxline.epfl.ch/RHEL/ws-5/update5/64-bit-x86/AMD64_iEM64T/images/boot.iso

Server i386: linuxline.epfl.ch/RHEL/as-5/update5/32-bit-x86/i386/images/boot.iso

x86_64: linuxline.epfl.ch/RHEL/as-5/update5/64-bit-x86/AMD64_iEM64T/images/boot.iso

A.1.3 Fedora 13

i386: linuxline.epfl.ch/fedora/i386/os/images/boot.iso

x86_64: linuxline.epfl.ch/Fedora/f13/Fedora/x86_64/os/images/boot.iso

A.1.4 openSUSE 11.3

À titre de démonstration, une distribution équivalente en qualité à Red Hat.

i586: mirror.switch.ch/ftp/mirror/opensuse/distribution/11.3/iso/openSUSE-11.3-NET-i586.iso

x86_64: mirror.switch.ch/ftp/mirror/opensuse/distribution/11.3/iso/openSUSE-11.3-NET-x86_64.iso

openSUSE Boot Options: en.opensuse.org/SDB:Linuxrc

A.2 Boot Options

Ubuntu Boot Options: <https://help.ubuntu.com/10.04/installation-guide/i386/preseed-contents.html>

Anaconda Boot Options: fedoraproject.org/wiki/Anaconda/Options

A.3 Isolinux

La configuration du fichier isolinux.cfg est identique à celle de syslinux.cfg.

ISOLINUX: syslinux.zytor.com/wiki/index.php/ISOLINUX

syslinux.cfg: syslinux.zytor.com/wiki/index.php/SYSLINUX

A.4 Installateur universel

Je cite au passage deux liens qui nous montrent que l'on verra peut-être un jour un installateur Linux universel.

Fedora: boot.fedoraproject.org/index

kernel.org: boot.kernel.org/

A.5 Dernier exemple

Image ISO: <https://documents.epfl.ch/users/b/ba/barras/www/Download/bootISO/myBoot.iso> ■